EXHIBIT 21

E.D. Tex. Case No. 2:24-CV-00064-JRG

Appendix B-24: Invalidity of U.S. Patent No. 7,784,058 in view of Solaris Zones

REFERENCE: Solaris Zones is a software application for implementing operating system-level virtualization technology. Solaris Zones anticipates and/or renders obvious, including under Plaintiff's apparent infringement theory,[1] all asserted claims[2] of U.S. Patent No. 7,784,058 (the "'058 patent") under 35 U.S.C. §§ 102 and/or 103. Solaris Zones was available at least by February 2004.[3] However, testimony from VirtaMove inventor Don Rochette indicates that a version of Solaris Zones disclosing the asserted claim limitations was available as early as 2002. *E.g.*, Rochette Dep. Tr. 54-57. This earlier date for Solaris Zones is further supported by the Tucker references described in Charts A-1, A-2, and A-3. Solaris Zones is prior art to the '058 patent under at least 35 U.S.C. § 102(b).

The chart below provides representative examples of where each element of each claim is found within Solaris Zones, including under Plaintiff's apparent construction of the asserted claims (and to the extent the claims are not found indefinite under 35 U.S.C. § 112). The cited evidence is merely illustrative, and Defendant reserves the right to cite alternative or additional evidence. To the extent Plaintiff contends that Solaris Zones does not disclose any asserted claims or claim elements of the '058 Patent, it would have been obvious to combine the teachings of Solaris Zones with: (1) the knowledge of one of ordinary skill in the art to show all the limitations of the claims; (2) the teachings of any of the prior art references set forth in Defendant's other invalidity charts with respect to the one or more limitations; and/or (3) the teachings of any of the prior art references set forth in the cover document of Defendant's Invalidity Contentions with respect to the '058 patent. Plaintiff has yet to identify any limitation of the asserted claims that it contends is not fully disclosed by Solaris Zones, either alone or in combination with other prior art cited by Defendant and/or with the knowledge of one of ordinary skill in the art. To the extent Plaintiff makes any such contention in the future, Defendant expressly reserves the right to rebut any such contention, including by identifying additional obviousness combinations.

---

[1] To the extent that these Invalidity Contentions rely on or otherwise embody particular constructions of terms or phrases in the Asserted Claims, Defendant is not proposing any such constructions as proper constructions of those terms or phrases. Various positions put forth in this document are predicated on Plaintiff's incorrect and overly broad interpretation of its claims as evidenced by its Infringement Contentions provided to Defendant. Those positions are not intended to and do not necessarily reflect Defendant's interpretation of the true and proper scope of Plaintiff's claims, and Defendant reserves the right to adopt claim construction positions that differ from or even conflict with various positions put forth in this document.

[2] As used herein, "asserted claims" refers only to those claims charted in Plaintiff's July 1, 2024, Infringement Contentions. To the extent Plaintiff later obtains leave to assert any additional claims for this patent, Defendant will provide its preliminary invalidity contentions consistent with the timing requirements set forth in the Court's order.

[3] Solaris Containers, https://en.wikipedia.org/wiki/Solaris_Containers.

Chart B-24

Invalidity Contentions: Solaris Zones

Where the chart below states that Solaris Zones "discloses" a limitation, such disclosure may be express or inherent. All emphasis is added unless otherwise indicated.

Fact Discovery is ongoing and Defendant's prior art investigation, including via third party discovery, is therefore not yet complete. Defendant reserves the right to rely upon additional evidence of invalidity obtained in the future as to Solaris Zones or any other prior art public use/sales/offers for sale that may anticipate or render obvious one or more asserted claims of the Asserted Patents under 35 U.S.C. § 102(b) and/or 35 U.S.C. § 103. Defendant may rely on additional documents and/or things that have not yet been located and/or testimony to support the contentions regarding Solaris Zones set forth in this chart.

Chart B-24 **Invalidity Contentions: Solaris Zones**

Claim 1

’058 Patent Claim 1	Disclosure
1[p] A computing system for executing a plurality of software applications comprising:	Solaris Zones, as evidenced by the example citations below, discloses a computing system for executing a plurality of software applications. *See, e.g.*: • Rochette Tr. at 37:1-14 (“Q. Okay. So what was your understanding of Solaris zones? A. Oh gosh. That is a lengthy answer. Zones would provide a separate container, a separate security environment for each app and for our multiple -- and for an application to run it. So you could define a zone for each application you had running in Solaris. That zone would allow the application to have it -- potentially its own version of files. It would -- I don't recall if it had isolated networking, but it would provide a secure environment, and it would provide a separate instance of files and so forth.”) • Rochette Tr. at 76:13-15 (“Q. And Solaris zones allowed applications to be executed in a secure environment? A. Correct.”) • Rochette Tr. at 90:2-92:5 ("Q. And do you know if the claims of the '058 patent reflect that difference? A. When I reviewed the patents in preparation for this discussion, I did not look at the claims. I don't -- I couldn't answer that effectively right now. . . . Q. And so, as you sit here today, are you able to identify any difference between the claims of the '058 patent and what was in Solaris zones? A. I would have a difficult time being specific about claims because, as I said, I didn't review the claims. I can -- I am comfortable with differences in technology or differences in implementation, but the specific claims I would have to -- I couldn't answer right now.") • Rochette Tr. at 83:22-88:8 ("Q. I think you mentioned earlier that Solaris zones had the capability of moving certain functionality that would normally be in an operating system kernel into a shared library; is that right? A. Yes, there were some capabilities that were represented outside of the kernel in user mode, correct. Q. What capabilities were those? A. File system separation. The use of something like the chroot command, or the chroot command you referred to earlier. The -- oh, I can't think of other examples. I mean, there were several examples of it. Q. And how did you learn that Solaris zones had that capability? A. Sitting down in front of a Solaris operating system and using it. Reading the documentations and using the capabilities and testing it. Q. And how did you determine that those capabilities were implemented in shared libraries rather than the kernel? A. If you go to an application and -- you would execute the command LDD. LDD shows you what libraries are required by a certain executable, and it would -- it's very clear -- clearly shown what libraries are referenced, and you could then look at the libraries and very clearly see what is implemented in those libraries. Q. So do you know if there was any shared library functionality in Solaris that was also in the kernel, in other words, the shared library functionality was a duplicate or performing a similar function to what was in the kernel? A. There were some networking capabilities and shared libraries and some -- for example, that would define what IP address is used, define

’058 Patent Claim 1	Disclosure
	what routing is used for a container separate from the kernel and was -- would be accessed by applications before going to the kernel. So you would have had a very similar capability in the kernel that was accessed by applications in these zones before going to the kernel. Q. So just for the sake of a concrete discussion, can you think of an example of a shared library that we could talk about that had this operating system functionality? A. I can't recall the names of them in Solaris. They would have been a lib.network.so or lib -- something with the name 'network' in it or 'net.' And it would have the network -- the things that would otherwise -- the things that were also present in a kernel, like defining an IP address, like defining routes, like defining -- not TCP protocol, but these would have been used by zones to configure the network capabilities present in that zone or container for use by the application. I don't remember the names. It's a long time ago, but there were shared libraries used by Solaris that did those things. Q. So for the networking shared library, if there were two applications using that shared library, would they each have their own instance in the sense of, you know, internal data used by the shared library having two different copies of that data? A. The way it works is the code for that library in question, once loaded into memory, is shared by all applications. That's the -- hence, the term 'shared library.' And -- so multiple zones could make use of the same networking library. The difference is that the data they use is distinct. So the actual code that gets executed is common across zones; the data that they use is separate for each zone. And any writes that would happen into memory . . . would implement an operating feature called copy-on-write, which would create a different instance of a page in memory, but it's specific to that zone.") • “Solaris has this wonderful new feature called "Zones". In addition to the many features that Solaris usually ships with we now have a new and very powerful feature that allows the administrator to create a virtual server within the server. Your server is no longer just a collection of user accounts and applications but it is now a collection of virtual servers within a global zone server. BSD Jail you say? A chroot environment? The comparison can be made to these older tools in much the same way that you can compare an aardvark to an elephant. They both walk and they both have four legs but one of them will carry you and a ton of cargo, the other, well, it just happens to have one vowel.” https://web.archive.org/web/20060613080532/http:/www.blastwave.org/articles/DMC-0002/index.html. • “Server consolidation, which allows multiple workloads to run on the same system, has become increasingly important as a way to improve the utilization of computing resources and reduce costs. Consolidation is common in mainframe environments, where technology to support running multiple workloads and even multiple operating systems on the same hardware has been evolving since the late 1960's. This technology is now becoming an important differentiator in the UNIX and Linux server market as well, both at the low end (virtual web hosting) and high end (traditional data center server consolidation). This paper introduces Solaris Zones (zones), a fully realized solution for server consolidation projects in a commercial UNIX operating system. By creating virtualized application execution environments within a single instance of the operating system, the facility strikes a unique balance between competing requirements. On the one hand, a system with multiple workloads needs to run

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 1	Disclosure
	those workloads in isolation, to ensure that applications can neither observe data from other applications nor affect their operation. It must also prevent applications from over-consuming system resources. On the other hand, the system as a whole has to be flexible, manageable, and observable, in order to reduce administrative costs and increase efficiency. By focusing on the support of multiple application environments rather than multiple operating system instances, zones meets isolation requirements without sacrificing manageability.” Price , D. and Tucker, A. (2004) *Solaris Zones: Operating System Support for Consolidating Commercial Workloads*. [https://www.researchgate.net/publication/220900659_Solaris_Zones_Operating_System_Support_for_Consolidating_Commercial_Workloads] • “Zones are a new operating system abstraction for partitioning systems, allowing multiple applications to run in isolation from each other on the same physical hardware. This isolation prevents processes running within a zone from monitoring or affecting processes running in other zones, seeing each other’s data, or manipulating the underlying hardware. Zones also provide an abstraction layer that separates applications from physical attributes of the machine on which they are deployed, such as physical device paths and network interface names.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • “These efforts differ from virtual machine implementations in that there is only one underlying operating system kernel, which is enhanced to provide increased isolation between groups of processes. The result is the ability to run multiple applications in isolation from each other within a single operating system instance.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][1], Chart A-2 (Tucker ‘556) at limitation 1[p][1], and Chart A-3 (Tucker Provisional) at limitation 1[p][1] as if fully set forth herein. To the extent Solaris Zones does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Solaris Zones. Further, one of ordinary skill in the art would have been motivated to modify Solaris Zones or combine it with any of the present prior art references found in Defendant’s Invalidity Contentions and any supplements thereto and the relevant section of charts for other prior art for the ’058 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Solaris Zones.

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 1	Disclosure
1[a] a) a processor;	Solaris Zones, as evidenced by the example citations below, discloses a processor. *See, e.g.*: • Rochette Tr. at 37:1-14 (“Q. Okay. So what was your understanding of Solaris zones? A. Oh gosh. That is a lengthy answer. Zones would provide a separate container, a separate security environment for each app and for our multiple -- and for an application to run it. So you could define a zone for each application you had running in Solaris. That zone would allow the application to have it -- potentially its own version of files. It would -- I don't recall if it had isolated networking, but it would provide a secure environment, and it would provide a separate instance of files and so forth.”) • Rochette Tr. at 90:2-92:5 ("Q. And do you know if the claims of the '058 patent reflect that difference? A. When I reviewed the patents in preparation for this discussion, I did not look at the claims. I don't -- I couldn't answer that effectively right now. . . . Q. And so, as you sit here today, are you able to identify any difference between the claims of the '058 patent and what was in Solaris zones? A. I would have a difficult time being specific about claims because, as I said, I didn't review the claims. I can -- I am comfortable with differences in technology or differences in implementation, but the specific claims I would have to -- I couldn't answer right now.") • Rochette Tr. at 83:22-88:8 ("Q. I think you mentioned earlier that Solaris zones had the capability of moving certain functionality that would normally be in an operating system kernel into a shared library; is that right? A. Yes, there were some capabilities that were represented outside of the kernel in user mode, correct. Q. What capabilities were those? A. File system separation. The use of something like the chroot command, or the chroot command you referred to earlier. The -- oh, I can't think of other examples. I mean, there were several examples of it. Q. And how did you learn that Solaris zones had that capability? A. Sitting down in front of a Solaris operating system and using it. Reading the documentations and using the capabilities and testing it. Q. And how did you determine that those capabilities were implemented in shared libraries rather than the kernel? A. If you go to an application and -- you would execute the command LDD. LDD shows you what libraries are required by a certain executable, and it would -- it's very clear -- clearly shown what libraries are referenced, and you could then look at the libraries and very clearly see what is implemented in those libraries. Q. So do you know if there was any shared library functionality in Solaris that was also in the kernel, in other words, the shared library functionality was a duplicate or performing a similar function to what was in the kernel? A. There were some networking capabilities and shared libraries and some -- for example, that would define what IP address is used, define what routing is used for a container separate from the kernel and was -- would be accessed by applications before going to the kernel. So you would have had a very similar capability in the kernel that was accessed by applications in these zones before going to the kernel. Q. So just for the sake of a concrete discussion, can you think of an example of a shared library that we could talk about that had this operating system functionality? A. I can't recall the names of them in Solaris. They would have been a lib.network.so or lib -- something with the name 'network' in it or 'net.' And it would have the network -- the things that

’058 Patent Claim 1	Disclosure
	would otherwise -- the things that were also present in a kernel, like defining an IP address, like defining routes, like defining -- not TCP protocol, but these would have been used by zones to configure the network capabilities present in that zone or container for use by the application. I don't remember the names. It's a long time ago, but there were shared libraries used by Solaris that did those things. Q. So for the networking shared library, if there were two applications using that shared library, would they each have their own instance in the sense of, you know, internal data used by the shared library having two different copies of that data? A. The way it works is the code for that library in question, once loaded into memory, is shared by all applications. That's the -- hence, the term 'shared library.' And -- so multiple zones could make use of the same networking library. The difference is that the data they use is distinct. So the actual code that gets executed is common across zones; the data that they use is separate for each zone. And any writes that would happen into memory . . . would implement an operating feature called copy-on-write, which would create a different instance of a page in memory, but it's specific to that zone.") • “Server consolidation, which allows multiple workloads to run on the same system, has become increasingly important as a way to improve the utilization of computing resources and reduce costs. Consolidation is common in mainframe environments, where technology to support running multiple workloads and even multiple operating systems on the same hardware has been evolving since the late 1960's. This technology is now becoming an important differentiator in the UNIX and Linux server market as well, both at the low end (virtual web hosting) and high end (traditional data center server consolidation). This paper introduces Solaris Zones (zones), a fully realized solution for server consolidation projects in a commercial UNIX operating system. By creating virtualized application execution environments within a single instance of the operating system, the facility strikes a unique balance between competing requirements. On the one hand, a system with multiple workloads needs to run those workloads in isolation, to ensure that applications can neither observe data from other applications nor affect their operation. It must also prevent applications from over-consuming system resources. On the other hand, the system as a whole has to be flexible, manageable, and observable, in order to reduce administrative costs and increase efficiency. By focusing on the support of multiple application environments rather than multiple operating system instances, zones meets isolation requirements without sacrificing manageability.” Price , D. and Tucker, A. (2004) *Solaris Zones: Operating System Support for Consolidating Commercial Workloads*. [https://www.researchgate.net/publication/220900659_ Solaris_Zones_Operating_System_Support_for_Consolidating_Commercial_Workloads] • “Zones are a new operating system abstraction for partitioning systems, allowing multiple applications to run in isolation from each other on the same physical hardware. This isolation prevents processes running within a zone from monitoring or affecting processes running in other zones, seeing each other’s data, or manipulating the underlying hardware. Zones also provide an abstraction layer that separates applications from physical attributes of the machine on which they are deployed, such as physical device paths and

’058 Patent Claim 1	Disclosure
	network interface names.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • “These efforts differ from virtual machine implementations in that there is only one underlying operating system kernel, which is enhanced to provide increased isolation between groups of processes. The result is the ability to run multiple applications in isolation from each other within a single operating system instance.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][1], Chart A-2 (Tucker ‘556) at limitation 1[p][1], and Chart A-3 (Tucker Provisional) at limitation 1[p][1] as if fully set forth herein. To the extent Solaris Zones does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Solaris Zones. Further, one of ordinary skill in the art would have been motivated to modify Solaris Zones or combine it with any of the present prior art references found in Defendant’s Invalidity Contentions and any supplements thereto and the relevant section of charts for other prior art for the ’058 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Solaris Zones.
1[b] b) an operating system having an operating system kernel having OS critical system elements (OSCSEs) for running in kernel mode using said processor; and,	Solaris Zones, as evidenced by the example citations below, discloses an operating system having an operating system kernel having OS critical system elements (OSCSEs) for running in kernel mode using said processor. *See, e.g.*: • Rochette Tr. at 26:3-8 (“Q. And you were aware of an operating system from Sun Microsystems called Solaris? A. Yes. Q. And Solaris predated the two patents that we’re looking at; correct? A. That’s correct.”) • Rochette Tr. at 27:3-20 (“Q. And are you aware of any examples, from your personal experience, of developers moving the functionality from an operating system’s kernel to a shared library? . . . A. A feature in the Sun Microsystems operating system called Solaris, the feature is called zones.”) • Rochette Tr. at 76:13-15 ("Q. And Solaris zones allowed applications to be executed in a secure environment? A. Correct.") • Rochette Tr. at 83:22-88:8 ("Q. I think you mentioned earlier that Solaris zones had the capability of moving certain functionality ***that would normally be in an operating system kernel into a shared library***; is that right? A. Yes, there were some capabilities that were represented outside of the kernel in user

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 1	Disclosure
	mode, correct. Q. What capabilities were those? A. File system separation. The use of something like the chroot command, or the chroot command you referred to earlier. The -- oh, I can't think of other examples. I mean, there were several examples of it. Q. And how did you learn that Solaris zones had that capability? A. Sitting down in front of a Solaris operating system and using it. Reading the documentations and using the capabilities and testing it. Q. And how did you determine that those capabilities were implemented in shared libraries rather than the kernel? A. If you go to an application and -- you would execute the command LDD. LDD shows you what libraries are required by a certain executable, and it would -- it's very clear -- clearly shown what libraries are referenced, and you could then look at the libraries and very clearly see what is implemented in those libraries. Q. So do you know if there was any shared library functionality in Solaris that was also in the kernel, in other words, the shared library functionality was a duplicate or performing a similar function to what was in the kernel? A. There were some networking capabilities and shared libraries and some -- for example, that would define what IP address is used, define what routing is used for a container separate from the kernel and was -- would be accessed by applications before going to the kernel. So you would have had a very similar capability in the kernel that was accessed by applications in these zones before going to the kernel. Q. So just for the sake of a concrete discussion, can you think of an example of a shared library that we could talk about that had this operating system functionality? A. I can't recall the names of them in Solaris. They would have been a lib.network.so or lib -- something with the name 'network' in it or 'net.' And it would have the network -- the things that would otherwise -- the things that were also present in a kernel, like defining an IP address, like defining routes, like defining -- not TCP protocol, but these would have been used by zones to configure the network capabilities present in that zone or container for use by the application. I don't remember the names. It's a long time ago, but there were shared libraries used by Solaris that did those things. Q. So for the networking shared library, if there were two applications using that shared library, would they each have their own instance in the sense of, you know, internal data used by the shared library having two different copies of that data? A. The way it works is the code for that library in question, once loaded into memory, is shared by all applications. That's the -- hence, the term 'shared library.' And -- so multiple zones could make use of the same networking library. The difference is that the data they use is distinct. So the actual code that gets executed is common across zones; the data that they use is separate for each zone. And any writes that would happen into memory . . . would implement an operating feature called copy-on-write, which would create a different instance of a page in memory, but it's specific to that zone.") • Rochette Tr. at 90:2-92:5 ("Q. And do you know if the claims of the '058 patent reflect that difference? A. When I reviewed the patents in preparation for this discussion, I did not look at the claims. I don't -- I couldn't answer that effectively right now. . . . Q. And so, as you sit here today, are you able to identify any difference between the claims of the '058 patent and what was in Solaris zones? A. I would have a difficult time being specific about claims because, as I said, I didn't review the claims. I can -- I am comfortable with differences in technology or differences in implementation, but the specific claims I would have to -- I couldn't answer right now.")

’058 Patent Claim 1	Disclosure
	• “Server consolidation, which allows multiple workloads to run on the same system, has become increasingly important as a way to improve the utilization of computing resources and reduce costs. Consolidation is common in mainframe environments, where technology to support running multiple workloads and even multiple operating systems on the same hardware has been evolving since the late 1960's. This technology is now becoming an important differentiator in the UNIX and Linux server market as well, both at the low end (virtual web hosting) and high end (traditional data center server consolidation). This paper introduces Solaris Zones (zones), a fully realized solution for server consolidation projects in a commercial UNIX operating system. By creating virtualized application execution environments within a single instance of the operating system, the facility strikes a unique balance between competing requirements. On the one hand, a system with multiple workloads needs to run those workloads in isolation, to ensure that applications can neither observe data from other applications nor affect their operation. It must also prevent applications from over-consuming system resources. On the other hand, the system as a whole has to be flexible, manageable, and observable, in order to reduce administrative costs and increase efficiency. By focusing on the support of multiple application environments rather than multiple operating system instances, zones meets isolation requirements without sacrificing manageability.” Price , D. and Tucker, A. (2004) *Solaris Zones: Operating System Support for Consolidating Commercial Workloads*. [https://www.researchgate.net/publication/220900659_Solaris_Zones_Operating_System_Support_for_Consolidating_Commercial_Workloads] • “Zones are a new operating system abstraction for partitioning systems, allowing multiple applications to run in isolation from each other on the same physical hardware. This isolation prevents processes running within a zone from monitoring or affecting processes running in other zones, seeing each other’s data, or manipulating the underlying hardware. Zones also provide an abstraction layer that separates applications from physical attributes of the machine on which they are deployed, such as physical device paths and network interface names.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • “These efforts differ from virtual machine implementations in that there is only one underlying operating system kernel, which is enhanced to provide increased isolation between groups of processes. The result is the ability to run multiple applications in isolation from each other within a single operating system instance.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][1], Chart A-2 (Tucker ‘556) at limitation 1[p][1], and Chart A-3 (Tucker Provisional) at limitation 1[p][1] as if fully set forth herein.

’058 Patent Claim 1	Disclosure
	To the extent Solaris Zones does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Solaris Zones. Further, one of ordinary skill in the art would have been motivated to modify Solaris Zones or combine it with any of the present prior art references found in Defendant’s Invalidity Contentions and any supplements thereto and the relevant section of charts for other prior art for the ’058 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Solaris Zones.
1[c] c) a shared library having shared library critical system elements (SLCSEs) stored therein for use by the plurality of software applications in user mode and	Solaris Zones, as evidenced by the example citations below, discloses a shared library having shared library critical system elements (SLCSEs) stored therein for use by the plurality of software applications in user mode. *See, e.g.,* the disclosures set forth at claim element 1[b]. *See also, e.g.*: • Rochette Tr. at 84:22-85:9 ("Q. I think you mentioned earlier that Solaris zones had the capability of moving certain functionality that would normally be in an operating system kernel into a shared library; is that right? A. ***Yes, there were some capabilities that were represented outside of the kernel in user mode, correct***.") • Rochette Tr. at 26:13-16 (“Q. Okay. And before – long before you filed these patent applications, Solaris had support for shared libraries; correct? A. Yes.”) • Rochette Tr. at 26:17-21 (“Q. And you were aware of Solaris’s support for shared libraries before you filed the applications that led to these two patents; correct? A. That is correct.”) • Rochette Tr. at 27:3-20 (“Q. And are you aware of any examples, from your personal experience, of developers moving the functionality from an operating system’s kernel to a shared library? . . . A. A feature in the Sun Microsystems operating system called Solaris, the feature is called zones.”) • Rochette Tr. at 63:2-8 ("Q. And by 2002 did software developers in Silicon Valley know how to move software functionality into a shared library? A. Yes.") • Rochette Tr. at 38:3-41:6 ("Q. So what are some of the differences between what Trigence AE was doing and what Solaris zones was doing? A. With zones, Solaris would utilize some operating system features to create a separate file area. They had modified the kernel, the internals of Solaris, to support this. It relied on kernel internals to provide separations between applications' end zones. It would provide separate scheduling types of mechanisms and so forth also implemented in the kernel. The Trigence solution made no changes to the kernel -- or didn't require any changes to the kernel. It was all done with what was called function overlays . . . It had nothing to do with separate zones or kernel changes or anything. Q. Okay. So when you say that Solaris zones was different from what Trigence was doing, you were contrasting Solaris zones with the Trigence product, not the patents; correct? A. Yes. Q. So can you explain these function overlays to me a little bit more. A. Function overlays use an operating system capability called library preload . . . So if the application called a function called fu, let's say, for example, if fu were existent in a system library and also in a Trigence library, because of the preload, the loader would resolve fu to the Trigence library instead of the system library because the Trigence library was

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 1	Disclosure
	loaded first due to LD_Preload/AppInit_DLL. Q. Okay. So you mentioned that this function overlay capability that Trigence was using was called LD_Preload in Solaris; is that right? A. Correct. Q. Okay. So Solaris had this LD_Preload capability that could be used to provide the function overlay capability? A. Correct. Q. And do you know when Solaris gained that function overlay capability that Trigence was using. A. That capability has been present in Solaris since its first release of its origin. Q. Does the 1987 sounds like approximately the right timeframe? A. Yeah, that would be -- that would be in the range, yes.") • Rochette Tr. at 83:22-88:8 ("Q. I think you mentioned earlier that Solaris zones had the capability of moving certain functionality that would normally be in an operating system kernel into a shared library; is that right? A. Yes, there were some capabilities that were represented outside of the kernel in user mode, correct. Q. What capabilities were those? A. File system separation. The use of something like the chroot command, or the chroot command you referred to earlier. The -- oh, I can't think of other examples. I mean, there were several examples of it. Q. And how did you learn that Solaris zones had that capability? A. Sitting down in front of a Solaris operating system and using it. Reading the documentations and using the capabilities and testing it. Q. And how did you determine that those capabilities were implemented in shared libraries rather than the kernel? A. If you go to an application and -- you would execute the command LDD. LDD shows you what libraries are required by a certain executable, and it would -- it's very clear -- clearly shown what libraries are referenced, and you could then look at the libraries and very clearly see what is implemented in those libraries. Q. So do you know if there was any shared library functionality in Solaris that was also in the kernel, in other words, the shared library functionality was a duplicate or performing a similar function to what was in the kernel? A. There were some networking capabilities and shared libraries and some -- for example, that would define what IP address is used, define what routing is used for a container separate from the kernel and was -- would be accessed by applications before going to the kernel. So you would have had a very similar capability in the kernel that was accessed by applications in these zones before going to the kernel. Q. So just for the sake of a concrete discussion, can you think of an example of a shared library that we could talk about that had this operating system functionality? A. I can't recall the names of them in Solaris. They would have been a lib.network.so or lib -- something with the name 'network' in it or 'net.' And it would have the network -- the things that would otherwise -- the things that were also present in a kernel, like defining an IP address, like defining routes, like defining -- not TCP protocol, but these would have been used by zones to configure the network capabilities present in that zone or container for use by the application. I don't remember the names. It's a long time ago, but there were shared libraries used by Solaris that did those things. Q. So for the networking shared library, if there were two applications using that shared library, would they each have their own instance in the sense of, you know, internal data used by the shared library having two different copies of that data? A. The way it works is the code for that library in question, once loaded into memory, is shared by all applications. That's the -- hence, the term 'shared library.' And -- so multiple zones could make use of the same networking library. The difference is that the data they use is distinct. So the actual code that gets executed is common across zones; the data that they use is separate for each

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 1	Disclosure
	zone. And any writes that would happen into memory . . . would implement an operating feature called copy-on-write, which would create a different instance of a page in memory, but it's specific to that zone.") • Rochette Tr. at 90:2-92:5 ("Q. And do you know if the claims of the '058 patent reflect that difference? A. When I reviewed the patents in preparation for this discussion, I did not look at the claims. I don't -- I couldn't answer that effectively right now. . . . Q. And so, as you sit here today, are you able to identify any difference between the claims of the '058 patent and what was in Solaris zones? A. I would have a difficult time being specific about claims because, as I said, I didn't review the claims. I can -- I am comfortable with differences in technology or differences in implementation, but the specific claims I would have to -- I couldn't answer right now.") • “Server consolidation, which allows multiple workloads to run on the same system, has become increasingly important as a way to improve the utilization of computing resources and reduce costs. Consolidation is common in mainframe environments, where technology to support running multiple workloads and even multiple operating systems on the same hardware has been evolving since the late 1960's. This technology is now becoming an important differentiator in the UNIX and Linux server market as well, both at the low end (virtual web hosting) and high end (traditional data center server consolidation). This paper introduces Solaris Zones (zones), a fully realized solution for server consolidation projects in a commercial UNIX operating system. By creating virtualized application execution environments within a single instance of the operating system, the facility strikes a unique balance between competing requirements. On the one hand, a system with multiple workloads needs to run those workloads in isolation, to ensure that applications can neither observe data from other applications nor affect their operation. It must also prevent applications from over-consuming system resources. On the other hand, the system as a whole has to be flexible, manageable, and observable, in order to reduce administrative costs and increase efficiency. By focusing on the support of multiple application environments rather than multiple operating system instances, zones meets isolation requirements without sacrificing manageability.” Price , D. and Tucker, A. (2004) *Solaris Zones: Operating System Support for Consolidating Commercial Workloads*. [https://www.researchgate.net/publication/220900659_Solaris_Zones_Operating_System_Support_for_Consolidating_Commercial_Workloads] • “Zones are a new operating system abstraction for partitioning systems, allowing multiple applications to run in isolation from each other on the same physical hardware. This isolation prevents processes running within a zone from monitoring or affecting processes running in other zones, seeing each other’s data, or manipulating the underlying hardware. Zones also provide an abstraction layer that separates applications from physical attributes of the machine on which they are deployed, such as physical device paths and network interface names.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.*

’058 Patent Claim 1	Disclosure
	• “These efforts differ from virtual machine implementations in that there is only one underlying operating system kernel, which is enhanced to provide increased isolation between groups of processes. The result is the ability to run multiple applications in isolation from each other within a single operating system instance.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][1], Chart A-2 (Tucker ‘556) at limitation 1[p][1], and Chart A-3 (Tucker Provisional) at limitation 1[p][1] as if fully set forth herein. To the extent Solaris Zones does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Solaris Zones. Further, one of ordinary skill in the art would have been motivated to modify Solaris Zones or combine it with any of the present prior art references found in Defendant’s Invalidity Contentions and any supplements thereto and the relevant section of charts for other prior art for the ’058 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Solaris Zones.
1[d][1] i) wherein some of the SLCSEs stored in the shared library are functional replicas of OSCSEs and are accessible to some of the plurality of software applications and	Solaris Zones, as evidenced by the example citations below, discloses that some of the SLCSEs stored in the shared library are functional replicas of OSCSEs and are accessible to some of the plurality of software applications. *See, e.g.,* the disclosures set forth at claim element 1[c]. *See also, e.g.*: • Rochette Tr. at 37:1-14 (“Q. Okay. So what was your understanding of Solaris zones? A. Oh gosh. That is a lengthy answer. Zones would provide a separate container, a separate security environment for each app and for our multiple -- and for an application to run it. So you could define a zone for each application you had running in Solaris. That zone would allow the application to have it -- potentially its own version of files. It would -- I don't recall if it had isolated networking, but it would provide a secure environment, and it would provide a separate instance of files and so forth.”) • Rochette Tr. at 63:2-8 ("Q. And by 2002 did software developers in Silicon Valley know how to move software functionality into a shared library? A. Yes.") • . Rochette Tr. at 83:22-88:8 ("Q. I think you mentioned earlier that Solaris zones had the capability of moving certain functionality that would normally be in an operating system kernel into a shared library; is that right? A. Yes, there were some capabilities that were represented outside of the kernel in user mode, correct. Q. What capabilities were those? A. File system separation. The use of something like the chroot command, or the chroot command you referred to earlier. The -- oh, I can't think of other examples. I

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 1	Disclosure
	mean, there were several examples of it. Q. And how did you learn that Solaris zones had that capability? A. Sitting down in front of a Solaris operating system and using it. Reading the documentations and using the capabilities and testing it. Q. And how did you determine that those capabilities were implemented in shared libraries rather than the kernel? A. If you go to an application and -- you would execute the command LDD. LDD shows you what libraries are required by a certain executable, and it would -- it's very clear -- clearly shown what libraries are referenced, and you could then look at the libraries and very clearly see what is implemented in those libraries. Q. So do you know if there was any shared library functionality in Solaris that was also in the kernel, in other words, the shared library functionality was a duplicate or performing a similar function to what was in the kernel? A. There were some networking capabilities and shared libraries and some -- for example, that would define what IP address is used, define what routing is used for a container separate from the kernel and was -- would be accessed by applications before going to the kernel. So you would have had a very similar capability in the kernel that was accessed by applications in these zones before going to the kernel. Q. So just for the sake of a concrete discussion, can you think of an example of a shared library that we could talk about that had this operating system functionality? A. I can't recall the names of them in Solaris. They would have been a lib.network.so or lib -- something with the name 'network' in it or 'net.' And it would have the network -- the things that would otherwise -- the things that were also present in a kernel, like defining an IP address, like defining routes, like defining -- not TCP protocol, but these would have been used by zones to configure the network capabilities present in that zone or container for use by the application. I don't remember the names. It's a long time ago, but there were shared libraries used by Solaris that did those things. Q. So for the networking shared library, if there were two applications using that shared library, would they each have their own instance in the sense of, you know, internal data used by the shared library having two different copies of that data? A. The way it works is the code for that library in question, once loaded into memory, is shared by all applications. That's the -- hence, the term 'shared library.' And -- so multiple zones could make use of the same networking library. The difference is that the data they use is distinct. So the actual code that gets executed is common across zones; the data that they use is separate for each zone. And any writes that would happen into memory . . . would implement an operating feature called copy-on-write, which would create a different instance of a page in memory, but it's specific to that zone.") • Rochette Tr. at 90:2-92:5 ("Q. And do you know if the claims of the '058 patent reflect that difference? A. When I reviewed the patents in preparation for this discussion, I did not look at the claims. I don't -- I couldn't answer that effectively right now. . . . Q. And so, as you sit here today, are you able to identify any difference between the claims of the '058 patent and what was in Solaris zones? A. I would have a difficult time being specific about claims because, as I said, I didn't review the claims. I can -- I am comfortable with differences in technology or differences in implementation, but the specific claims I would have to -- I couldn't answer right now.") • “Server consolidation, which allows multiple workloads to run on the same system, has become increasingly important as a way to improve the utilization of computing resources and reduce costs.

’058 Patent Claim 1	Disclosure
	Consolidation is common in mainframe environments, where technology to support running multiple workloads and even multiple operating systems on the same hardware has been evolving since the late 1960's. This technology is now becoming an important differentiator in the UNIX and Linux server market as well, both at the low end (virtual web hosting) and high end (traditional data center server consolidation). This paper introduces Solaris Zones (zones), a fully realized solution for server consolidation projects in a commercial UNIX operating system. By creating virtualized application execution environments within a single instance of the operating system, the facility strikes a unique balance between competing requirements. On the one hand, a system with multiple workloads needs to run those workloads in isolation, to ensure that applications can neither observe data from other applications nor affect their operation. It must also prevent applications from over-consuming system resources. On the other hand, the system as a whole has to be flexible, manageable, and observable, in order to reduce administrative costs and increase efficiency. By focusing on the support of multiple application environments rather than multiple operating system instances, zones meets isolation requirements without sacrificing manageability.” Price , D. and Tucker, A. (2004) *Solaris Zones: Operating System Support for Consolidating Commercial Workloads*. [https://www.researchgate.net/publication/220900659_Solaris_Zones_Operating_System_Support_for_Consolidating_Commercial_Workloads] • “Zones are a new operating system abstraction for partitioning systems, allowing multiple applications to run in isolation from each other on the same physical hardware. This isolation prevents processes running within a zone from monitoring or affecting processes running in other zones, seeing each other’s data, or manipulating the underlying hardware. Zones also provide an abstraction layer that separates applications from physical attributes of the machine on which they are deployed, such as physical device paths and network interface names.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • “These efforts differ from virtual machine implementations in that there is only one underlying operating system kernel, which is enhanced to provide increased isolation between groups of processes. The result is the ability to run multiple applications in isolation from each other within a single operating system instance.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][1], Chart A-2 (Tucker ‘556) at limitation 1[p][1], and Chart A-3 (Tucker Provisional) at limitation 1[p][1] as if fully set forth herein. To the extent Solaris Zones does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the

’058 Patent Claim 1	Disclosure
	teachings of Solaris Zones. Further, one of ordinary skill in the art would have been motivated to modify Solaris Zones or combine it with any of the present prior art references found in Defendant’s Invalidity Contentions and any supplements thereto and the relevant section of charts for other prior art for the ’058 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Solaris Zones.
1[d][2] when one of the SLCSEs is accessed by one or more of the plurality of software applications it forms a part of the one or more of the plurality of software applications,	Solaris Zones, as evidenced by the example citations below, discloses when one of the SLCSEs is accessed by one or more of the plurality of software applications it forms a part of the one or more of the plurality of software applications. *See, e.g.,* the disclosures set forth at claim elements 1[c] and 1[d][1]. *See also, e.g.*: • Rochette Tr. at 37:1-14 (“Q. Okay. So what was your understanding of Solaris zones? A. Oh gosh. That is a lengthy answer. Zones would provide a separate container, a separate security environment for each app and for our multiple -- and for an application to run it. So you could define a zone for each application you had running in Solaris. That zone would allow the application to have it -- potentially its own version of files. It would -- I don't recall if it had isolated networking, but it would provide a secure environment, and it would provide a separate instance of files and so forth.”) • Rochette Tr. at 83:22-88:8 ("Q. I think you mentioned earlier that Solaris zones had the capability of moving certain functionality that would normally be in an operating system kernel into a shared library; is that right? A. Yes, there were some capabilities that were represented outside of the kernel in user mode, correct. Q. What capabilities were those? A. File system separation. The use of something like the chroot command, or the chroot command you referred to earlier. The -- oh, I can't think of other examples. I mean, there were several examples of it. Q. And how did you learn that Solaris zones had that capability? A. Sitting down in front of a Solaris operating system and using it. Reading the documentations and using the capabilities and testing it. Q. And how did you determine that those capabilities were implemented in shared libraries rather than the kernel? A. If you go to an application and -- you would execute the command LDD. LDD shows you what libraries are required by a certain executable, and it would -- it's very clear -- clearly shown what libraries are referenced, and you could then look at the libraries and very clearly see what is implemented in those libraries. Q. So do you know if there was any shared library functionality in Solaris that was also in the kernel, in other words, the shared library functionality was a duplicate or performing a similar function to what was in the kernel? A. There were some networking capabilities and shared libraries and some -- for example, that would define what IP address is used, define what routing is used for a container separate from the kernel and was -- would be accessed by applications before going to the kernel. So you would have had a very similar capability in the kernel that was accessed by applications in these zones before going to the kernel. Q. So just for the sake of a concrete discussion, can you think of an example of a shared library that we could talk about that had this operating system functionality? A. I can't recall the names of them in Solaris. They would have been a lib.network.so or

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 1	Disclosure
	lib -- something with the name 'network' in it or 'net.' And it would have the network -- the things that would otherwise -- the things that were also present in a kernel, like defining an IP address, like defining routes, like defining -- not TCP protocol, but these would have been used by zones to configure the network capabilities present in that zone or container for use by the application. I don't remember the names. It's a long time ago, but there were shared libraries used by Solaris that did those things. Q. So for the networking shared library, if there were two applications using that shared library, would they each have their own instance in the sense of, you know, internal data used by the shared library having two different copies of that data? A. The way it works is the code for that library in question, once loaded into memory, is shared by all applications. That's the -- hence, the term 'shared library.' And -- so multiple zones could make use of the same networking library. The difference is that the data they use is distinct. So the actual code that gets executed is common across zones; the data that they use is separate for each zone. And any writes that would happen into memory . . . would implement an operating feature called copy-on-write, which would create a different instance of a page in memory, but it's specific to that zone.") • Rochette Tr. at 90:2-92:5 ("Q. And do you know if the claims of the '058 patent reflect that difference? A. When I reviewed the patents in preparation for this discussion, I did not look at the claims. I don't -- I couldn't answer that effectively right now. . . . Q. And so, as you sit here today, are you able to identify any difference between the claims of the '058 patent and what was in Solaris zones? A. I would have a difficult time being specific about claims because, as I said, I didn't review the claims. I can -- I am comfortable with differences in technology or differences in implementation, but the specific claims I would have to -- I couldn't answer right now.") • “Server consolidation, which allows multiple workloads to run on the same system, has become increasingly important as a way to improve the utilization of computing resources and reduce costs. Consolidation is common in mainframe environments, where technology to support running multiple workloads and even multiple operating systems on the same hardware has been evolving since the late 1960's. This technology is now becoming an important differentiator in the UNIX and Linux server market as well, both at the low end (virtual web hosting) and high end (traditional data center server consolidation). This paper introduces Solaris Zones (zones), a fully realized solution for server consolidation projects in a commercial UNIX operating system. By creating virtualized application execution environments within a single instance of the operating system, the facility strikes a unique balance between competing requirements. On the one hand, a system with multiple workloads needs to run those workloads in isolation, to ensure that applications can neither observe data from other applications nor affect their operation. It must also prevent applications from over-consuming system resources. On the other hand, the system as a whole has to be flexible, manageable, and observable, in order to reduce administrative costs and increase efficiency. By focusing on the support of multiple application environments rather than multiple operating system instances, zones meets isolation requirements without sacrificing manageability.”

’058 Patent Claim 1	Disclosure
	Price , D. and Tucker, A. (2004) *Solaris Zones: Operating System Support for Consolidating Commercial Workloads*. [https://www.researchgate.net/publication/220900659_ Solaris_Zones_Operating_System_Support_for_Consolidating_Commercial_Workloads] • “Zones are a new operating system abstraction for partitioning systems, allowing multiple applications to run in isolation from each other on the same physical hardware. This isolation prevents processes running within a zone from monitoring or affecting processes running in other zones, seeing each other’s data, or manipulating the underlying hardware. Zones also provide an abstraction layer that separates applications from physical attributes of the machine on which they are deployed, such as physical device paths and network interface names.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • “These efforts differ from virtual machine implementations in that there is only one underlying operating system kernel, which is enhanced to provide increased isolation between groups of processes. The result is the ability to run multiple applications in isolation from each other within a single operating system instance.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][1], Chart A-2 (Tucker ‘556) at limitation 1[p][1], and Chart A-3 (Tucker Provisional) at limitation 1[p][1] as if fully set forth herein. To the extent Solaris Zones does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Solaris Zones. Further, one of ordinary skill in the art would have been motivated to modify Solaris Zones or combine it with any of the present prior art references found in Defendant’s Invalidity Contentions and any supplements thereto and the relevant section of charts for other prior art for the ’058 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Solaris Zones.
1[e][1] ii) wherein an instance of a SLCSE provided to at least a first of the plurality of software applications from the shared library is run in a context of said at least first of the plurality of software applications without being shared with other of the plurality of software applications and	Solaris Zones, as evidenced by the example citations below, discloses wherein an instance of a SLCSE provided to at least a first of the plurality of software applications from the shared library is run in a context of said at least first of the plurality of software applications without being shared with other of the plurality of software applications. *See, e.g.,* the disclosures set forth at claim elements 1[c] and 1[d][1]-[2]. *See also, e.g.*: • Rochette Tr. at 37:1-14 (“Q. Okay. So what was your understanding of Solaris zones? A. Oh gosh. That is a lengthy answer. Zones would provide a separate container, a separate security environment for each app and for our multiple -- and for an application to run it. So you could define a zone for each application you

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 1	Disclosure
	had running in Solaris. That zone would allow the application to have it -- potentially its own version of files. It would -- I don't recall if it had isolated networking, but it would provide a secure environment, and it would provide a separate instance of files and so forth.”) • Rochette Tr. at 83:22-88:8 ("Q. I think you mentioned earlier that Solaris zones had the capability of moving certain functionality that would normally be in an operating system kernel into a shared library; is that right? A. Yes, there were some capabilities that were represented outside of the kernel in user mode, correct. Q. What capabilities were those? A. File system separation. The use of something like the chroot command, or the chroot command you referred to earlier. The -- oh, I can't think of other examples. I mean, there were several examples of it. Q. And how did you learn that Solaris zones had that capability? A. Sitting down in front of a Solaris operating system and using it. Reading the documentations and using the capabilities and testing it. Q. And how did you determine that those capabilities were implemented in shared libraries rather than the kernel? A. If you go to an application and -- you would execute the command LDD. LDD shows you what libraries are required by a certain executable, and it would -- it's very clear -- clearly shown what libraries are referenced, and you could then look at the libraries and very clearly see what is implemented in those libraries. Q. So do you know if there was any shared library functionality in Solaris that was also in the kernel, in other words, the shared library functionality was a duplicate or performing a similar function to what was in the kernel? A. There were some networking capabilities and shared libraries and some -- for example, that would define what IP address is used, define what routing is used for a container separate from the kernel and was -- would be accessed by applications before going to the kernel. So you would have had a very similar capability in the kernel that was accessed by applications in these zones before going to the kernel. Q. So just for the sake of a concrete discussion, can you think of an example of a shared library that we could talk about that had this operating system functionality? A. I can't recall the names of them in Solaris. They would have been a lib.network.so or lib -- something with the name 'network' in it or 'net.' And it would have the network -- the things that would otherwise -- the things that were also present in a kernel, like defining an IP address, like defining routes, like defining -- not TCP protocol, but these would have been used by zones to configure the network capabilities present in that zone or container for use by the application. I don't remember the names. It's a long time ago, but there were shared libraries used by Solaris that did those things. Q. So for the networking shared library, if there were two applications using that shared library, would they each have their own instance in the sense of, you know, internal data used by the shared library having two different copies of that data? A. The way it works is the code for that library in question, once loaded into memory, is shared by all applications. That's the -- hence, the term 'shared library.' And -- so multiple zones could make use of the same networking library. The difference is that the data they use is distinct. So the actual code that gets executed is common across zones; the data that they use is separate for each zone. And any writes that would happen into memory . . . would implement an operating feature called copy-on-write, which would create a different instance of a page in memory, but it's specific to that zone.")

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 1	Disclosure
	• Rochette Tr. at 90:2-92:5 ("Q. And do you know if the claims of the '058 patent reflect that difference? A. When I reviewed the patents in preparation for this discussion, I did not look at the claims. I don't -- I couldn't answer that effectively right now. . . . Q. And so, as you sit here today, are you able to identify any difference between the claims of the '058 patent and what was in Solaris zones? A. I would have a difficult time being specific about claims because, as I said, I didn't review the claims. I can -- I am comfortable with differences in technology or differences in implementation, but the specific claims I would have to -- I couldn't answer right now.") • “Server consolidation, which allows multiple workloads to run on the same system, has become increasingly important as a way to improve the utilization of computing resources and reduce costs. Consolidation is common in mainframe environments, where technology to support running multiple workloads and even multiple operating systems on the same hardware has been evolving since the late 1960's. This technology is now becoming an important differentiator in the UNIX and Linux server market as well, both at the low end (virtual web hosting) and high end (traditional data center server consolidation). This paper introduces Solaris Zones (zones), a fully realized solution for server consolidation projects in a commercial UNIX operating system. By creating virtualized application execution environments within a single instance of the operating system, the facility strikes a unique balance between competing requirements. On the one hand, a system with multiple workloads needs to run those workloads in isolation, to ensure that applications can neither observe data from other applications nor affect their operation. It must also prevent applications from over-consuming system resources. On the other hand, the system as a whole has to be flexible, manageable, and observable, in order to reduce administrative costs and increase efficiency. By focusing on the support of multiple application environments rather than multiple operating system instances, zones meets isolation requirements without sacrificing manageability.” Price , D. and Tucker, A. (2004) *Solaris Zones: Operating System Support for Consolidating Commercial Workloads*. [https://www.researchgate.net/publication/220900659_Solaris_Zones_Operating_System_Support_for_Consolidating_Commercial_Workloads] • “Zones are a new operating system abstraction for partitioning systems, allowing multiple applications to run in isolation from each other on the same physical hardware. This isolation prevents processes running within a zone from monitoring or affecting processes running in other zones, seeing each other’s data, or manipulating the underlying hardware. Zones also provide an abstraction layer that separates applications from physical attributes of the machine on which they are deployed, such as physical device paths and network interface names.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • “These efforts differ from virtual machine implementations in that there is only one underlying operating system kernel, which is enhanced to provide increased isolation between groups of processes. The result is the ability to run multiple applications in isolation from each other within a single operating system instance.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.*

’058 Patent Claim 1	Disclosure
	Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][1], Chart A-2 (Tucker ‘556) at limitation 1[p][1], and Chart A-3 (Tucker Provisional) at limitation 1[p][1] as if fully set forth herein. To the extent Solaris Zones does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Solaris Zones. Further, one of ordinary skill in the art would have been motivated to modify Solaris Zones or combine it with any of the present prior art references found in Defendant’s Invalidity Contentions and any supplements thereto and the relevant section of charts for other prior art for the ’058 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Solaris Zones.
1[e][2] where at least a second of the plurality of software applications running under the operating system have use of a unique instance of a corresponding critical system element for performing same function, and	Solaris Zones, as evidenced by the example citations below, discloses where at least a second of the plurality of software applications running under the operating system have use of a unique instance of a corresponding critical system element for performing same function. *See, e.g.,* the disclosures set forth at claim elements 1[c] and 1[d][1]-[2]. *See also, e.g.*: • Rochette Tr. at 37:1-14 (“Q. Okay. So what was your understanding of Solaris zones? A. Oh gosh. That is a lengthy answer. Zones would provide a separate container, a separate security environment for each app and for our multiple -- and for an application to run it. So you could define a zone for each application you had running in Solaris. That zone would allow the application to have it -- potentially its own version of files. It would -- I don't recall if it had isolated networking, but it would provide a secure environment, and it would provide a separate instance of files and so forth.”) • Rochette Tr. at 83:22-88:8 ("Q. I think you mentioned earlier that Solaris zones had the capability of moving certain functionality that would normally be in an operating system kernel into a shared library; is that right? A. Yes, there were some capabilities that were represented outside of the kernel in user mode, correct. Q. What capabilities were those? A. File system separation. The use of something like the chroot command, or the chroot command you referred to earlier. The -- oh, I can't think of other examples. I mean, there were several examples of it. Q. And how did you learn that Solaris zones had that capability? A. Sitting down in front of a Solaris operating system and using it. Reading the documentations and using the capabilities and testing it. Q. And how did you determine that those capabilities were implemented in shared libraries rather than the kernel? A. If you go to an application and -- you would execute the command LDD. LDD shows you what libraries are required by a certain executable, and it would -- it's

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 1	Disclosure
	very clear -- clearly shown what libraries are referenced, and you could then look at the libraries and very clearly see what is implemented in those libraries. Q. So do you know if there was any shared library functionality in Solaris that was also in the kernel, in other words, the shared library functionality was a duplicate or performing a similar function to what was in the kernel? A. There were some networking capabilities and shared libraries and some -- for example, that would define what IP address is used, define what routing is used for a container separate from the kernel and was -- would be accessed by applications before going to the kernel. So you would have had a very similar capability in the kernel that was accessed by applications in these zones before going to the kernel. Q. So just for the sake of a concrete discussion, can you think of an example of a shared library that we could talk about that had this operating system functionality? A. I can't recall the names of them in Solaris. They would have been a lib.network.so or lib -- something with the name 'network' in it or 'net.' And it would have the network -- the things that would otherwise -- the things that were also present in a kernel, like defining an IP address, like defining routes, like defining -- not TCP protocol, but these would have been used by zones to configure the network capabilities present in that zone or container for use by the application. I don't remember the names. It's a long time ago, but there were shared libraries used by Solaris that did those things. Q. So for the networking shared library, if there were two applications using that shared library, would they each have their own instance in the sense of, you know, internal data used by the shared library having two different copies of that data? A. The way it works is the code for that library in question, once loaded into memory, is shared by all applications. That's the -- hence, the term 'shared library.' And -- so multiple zones could make use of the same networking library. The difference is that the data they use is distinct. So the actual code that gets executed is common across zones; the data that they use is separate for each zone. And any writes that would happen into memory . . . would implement an operating feature called copy-on-write, which would create a different instance of a page in memory, but it's specific to that zone.") • Rochette Tr. at 90:2-92:5 ("Q. And do you know if the claims of the '058 patent reflect that difference? A. When I reviewed the patents in preparation for this discussion, I did not look at the claims. I don't -- I couldn't answer that effectively right now. . . . Q. And so, as you sit here today, are you able to identify any difference between the claims of the '058 patent and what was in Solaris zones? A. I would have a difficult time being specific about claims because, as I said, I didn't review the claims. I can -- I am comfortable with differences in technology or differences in implementation, but the specific claims I would have to -- I couldn't answer right now.") • “Server consolidation, which allows multiple workloads to run on the same system, has become increasingly important as a way to improve the utilization of computing resources and reduce costs. Consolidation is common in mainframe environments, where technology to support running multiple workloads and even multiple operating systems on the same hardware has been evolving since the late 1960's. This technology is now becoming an important differentiator in the UNIX and Linux server market as well, both at the low end (virtual web hosting) and high end (traditional data center server consolidation). This paper introduces Solaris Zones (zones), a fully realized solution for server

’058 Patent Claim 1	Disclosure
	consolidation projects in a commercial UNIX operating system. By creating virtualized application execution environments within a single instance of the operating system, the facility strikes a unique balance between competing requirements. On the one hand, a system with multiple workloads needs to run those workloads in isolation, to ensure that applications can neither observe data from other applications nor affect their operation. It must also prevent applications from over-consuming system resources. On the other hand, the system as a whole has to be flexible, manageable, and observable, in order to reduce administrative costs and increase efficiency. By focusing on the support of multiple application environments rather than multiple operating system instances, zones meets isolation requirements without sacrificing manageability.” Price , D. and Tucker, A. (2004) *Solaris Zones: Operating System Support for Consolidating Commercial Workloads*. [https://www.researchgate.net/publication/220900659_ Solaris_Zones_Operating_System_Support_for_Consolidating_Commercial_Workloads] • “Zones are a new operating system abstraction for partitioning systems, allowing multiple applications to run in isolation from each other on the same physical hardware. This isolation prevents processes running within a zone from monitoring or affecting processes running in other zones, seeing each other’s data, or manipulating the underlying hardware. Zones also provide an abstraction layer that separates applications from physical attributes of the machine on which they are deployed, such as physical device paths and network interface names.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • “These efforts differ from virtual machine implementations in that there is only one underlying operating system kernel, which is enhanced to provide increased isolation between groups of processes. The result is the ability to run multiple applications in isolation from each other within a single operating system instance.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][1], Chart A-2 (Tucker ‘556) at limitation 1[p][1], and Chart A-3 (Tucker Provisional) at limitation 1[p][1] as if fully set forth herein. To the extent Solaris Zones does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Solaris Zones. Further, one of ordinary skill in the art would have been motivated to modify Solaris Zones or combine it with any of the present prior art references found in Defendant’s Invalidity Contentions and any supplements thereto and the relevant section of charts for other prior art for the ’058 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Solaris Zones.

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 1	Disclosure
1[f][1] iii) wherein a SLCSE related to a predetermined function is provided to the first of the plurality of software applications for running a first instance of the SLCSE, and	Solaris Zones, as evidenced by the example citations below, discloses wherein a SLCSE related to a predetermined function is provided to the first of the plurality of software applications for running a first instance of the SLCSE. *See, e.g.,* the disclosures set forth at claim elements 1[c] and 1[d][1]-[2]. *See also, e.g.*: • Rochette Tr. at 37:1-14 (“Q. Okay. So what was your understanding of Solaris zones? A. Oh gosh. That is a lengthy answer. Zones would provide a separate container, a separate security environment for each app and for our multiple -- and for an application to run it. So you could define a zone for each application you had running in Solaris. That zone would allow the application to have it -- potentially its own version of files. It would -- I don't recall if it had isolated networking, but it would provide a secure environment, and it would provide a separate instance of files and so forth.”) • Rochette Tr. at 83:22-88:8 ("Q. I think you mentioned earlier that Solaris zones had the capability of moving certain functionality that would normally be in an operating system kernel into a shared library; is that right? A. Yes, there were some capabilities that were represented outside of the kernel in user mode, correct. Q. What capabilities were those? A. File system separation. The use of something like the chroot command, or the chroot command you referred to earlier. The -- oh, I can't think of other examples. I mean, there were several examples of it. Q. And how did you learn that Solaris zones had that capability? A. Sitting down in front of a Solaris operating system and using it. Reading the documentations and using the capabilities and testing it. Q. And how did you determine that those capabilities were implemented in shared libraries rather than the kernel? A. If you go to an application and -- you would execute the command LDD. LDD shows you what libraries are required by a certain executable, and it would -- it's very clear -- clearly shown what libraries are referenced, and you could then look at the libraries and very clearly see what is implemented in those libraries. Q. So do you know if there was any shared library functionality in Solaris that was also in the kernel, in other words, the shared library functionality was a duplicate or performing a similar function to what was in the kernel? A. There were some networking capabilities and shared libraries and some -- for example, that would define what IP address is used, define what routing is used for a container separate from the kernel and was -- would be accessed by applications before going to the kernel. So you would have had a very similar capability in the kernel that was accessed by applications in these zones before going to the kernel. Q. So just for the sake of a concrete discussion, can you think of an example of a shared library that we could talk about that had this operating system functionality? A. I can't recall the names of them in Solaris. They would have been a lib.network.so or lib -- something with the name 'network' in it or 'net.' And it would have the network -- the things that would otherwise -- the things that were also present in a kernel, like defining an IP address, like defining routes, like defining -- not TCP protocol, but these would have been used by zones to configure the network capabilities present in that zone or container for use by the application. I don't remember the names. It's a long time ago, but there were shared libraries used by Solaris that did those things. Q. So for the networking shared library, if there were two applications using that shared library, would they each have their own instance in the sense of, you know, internal data used by the shared library having two

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 1	Disclosure
	different copies of that data? A. The way it works is the code for that library in question, once loaded into memory, is shared by all applications. That's the -- hence, the term 'shared library.' And -- so multiple zones could make use of the same networking library. The difference is that the data they use is distinct. So the actual code that gets executed is common across zones; the data that they use is separate for each zone. And any writes that would happen into memory . . . would implement an operating feature called copy-on-write, which would create a different instance of a page in memory, but it's specific to that zone.") • Rochette Tr. at 90:2-92:5 ("Q. And do you know if the claims of the '058 patent reflect that difference? A. When I reviewed the patents in preparation for this discussion, I did not look at the claims. I don't -- I couldn't answer that effectively right now. . . . Q. And so, as you sit here today, are you able to identify any difference between the claims of the '058 patent and what was in Solaris zones? A. I would have a difficult time being specific about claims because, as I said, I didn't review the claims. I can -- I am comfortable with differences in technology or differences in implementation, but the specific claims I would have to -- I couldn't answer right now.") • “Server consolidation, which allows multiple workloads to run on the same system, has become increasingly important as a way to improve the utilization of computing resources and reduce costs. Consolidation is common in mainframe environments, where technology to support running multiple workloads and even multiple operating systems on the same hardware has been evolving since the late 1960's. This technology is now becoming an important differentiator in the UNIX and Linux server market as well, both at the low end (virtual web hosting) and high end (traditional data center server consolidation). This paper introduces Solaris Zones (zones), a fully realized solution for server consolidation projects in a commercial UNIX operating system. By creating virtualized application execution environments within a single instance of the operating system, the facility strikes a unique balance between competing requirements. On the one hand, a system with multiple workloads needs to run those workloads in isolation, to ensure that applications can neither observe data from other applications nor affect their operation. It must also prevent applications from over-consuming system resources. On the other hand, the system as a whole has to be flexible, manageable, and observable, in order to reduce administrative costs and increase efficiency. By focusing on the support of multiple application environments rather than multiple operating system instances, zones meets isolation requirements without sacrificing manageability.” Price , D. and Tucker, A. (2004) *Solaris Zones: Operating System Support for Consolidating Commercial Workloads*. [https://www.researchgate.net/publication/220900659_Solaris_Zones_Operating_System_Support_for_Consolidating_Commercial_Workloads] • “Zones are a new operating system abstraction for partitioning systems, allowing multiple applications to run in isolation from each other on the same physical hardware. This isolation prevents processes running within a zone from monitoring or affecting processes running in other zones, seeing each other’s data, or manipulating the underlying hardware. Zones also provide an abstraction layer that separates applications

’058 Patent Claim 1	Disclosure
	from physical attributes of the machine on which they are deployed, such as physical device paths and network interface names.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • “These efforts differ from virtual machine implementations in that there is only one underlying operating system kernel, which is enhanced to provide increased isolation between groups of processes. The result is the ability to run multiple applications in isolation from each other within a single operating system instance.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][1], Chart A-2 (Tucker ‘556) at limitation 1[p][1], and Chart A-3 (Tucker Provisional) at limitation 1[p][1] as if fully set forth herein. To the extent Solaris Zones does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Solaris Zones. Further, one of ordinary skill in the art would have been motivated to modify Solaris Zones or combine it with any of the present prior art references found in Defendant’s Invalidity Contentions and any supplements thereto and the relevant section of charts for other prior art for the ’058 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Solaris Zones.
1[f][2] wherein a SLCSE for performing a same function is provided to the second of the plurality of software applications for running a second instance of the SLCSE simultaneously.	Solaris Zones, as evidenced by the example citations below, discloses wherein a SLCSE for performing a same function is provided to the second of the plurality of software applications for running a second instance of the SLCSE simultaneously. *See, e.g.,* the disclosures set forth at claim elements 1[c] and 1[d][1]-[2]. *See also, e.g.*: • Rochette Tr. at 37:1-14 (“Q. Okay. So what was your understanding of Solaris zones? A. Oh gosh. That is a lengthy answer. Zones would provide a separate container, a separate security environment for each app and for our multiple -- and for an application to run it. So you could define a zone for each application you had running in Solaris. That zone would allow the application to have it -- potentially its own version of files. It would -- I don't recall if it had isolated networking, but it would provide a secure environment, and it would provide a separate instance of files and so forth.”) • Rochette Tr. at 83:22-88:8 ("Q. I think you mentioned earlier that Solaris zones had the capability of moving certain functionality that would normally be in an operating system kernel into a shared library; is that right? A. Yes, there were some capabilities that were represented outside of the kernel in user mode, correct. Q. What capabilities were those? A. File system separation. The use of something like the chroot

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 1	Disclosure
	command, or the chroot command you referred to earlier. The -- oh, I can't think of other examples. I mean, there were several examples of it. Q. And how did you learn that Solaris zones had that capability? A. Sitting down in front of a Solaris operating system and using it. Reading the documentations and using the capabilities and testing it. Q. And how did you determine that those capabilities were implemented in shared libraries rather than the kernel? A. If you go to an application and -- you would execute the command LDD. LDD shows you what libraries are required by a certain executable, and it would -- it's very clear -- clearly shown what libraries are referenced, and you could then look at the libraries and very clearly see what is implemented in those libraries. Q. So do you know if there was any shared library functionality in Solaris that was also in the kernel, in other words, the shared library functionality was a duplicate or performing a similar function to what was in the kernel? A. There were some networking capabilities and shared libraries and some -- for example, that would define what IP address is used, define what routing is used for a container separate from the kernel and was -- would be accessed by applications before going to the kernel. So you would have had a very similar capability in the kernel that was accessed by applications in these zones before going to the kernel. Q. So just for the sake of a concrete discussion, can you think of an example of a shared library that we could talk about that had this operating system functionality? A. I can't recall the names of them in Solaris. They would have been a lib.network.so or lib -- something with the name 'network' in it or 'net.' And it would have the network -- the things that would otherwise -- the things that were also present in a kernel, like defining an IP address, like defining routes, like defining -- not TCP protocol, but these would have been used by zones to configure the network capabilities present in that zone or container for use by the application. I don't remember the names. It's a long time ago, but there were shared libraries used by Solaris that did those things. Q. So for the networking shared library, if there were two applications using that shared library, would they each have their own instance in the sense of, you know, internal data used by the shared library having two different copies of that data? A. The way it works is the code for that library in question, once loaded into memory, is shared by all applications. That's the -- hence, the term 'shared library.' And -- so multiple zones could make use of the same networking library. The difference is that the data they use is distinct. So the actual code that gets executed is common across zones; the data that they use is separate for each zone. And any writes that would happen into memory . . . would implement an operating feature called copy-on-write, which would create a different instance of a page in memory, but it's specific to that zone.") • Rochette Tr. at 90:2-92:5 ("Q. And do you know if the claims of the '058 patent reflect that difference? A. When I reviewed the patents in preparation for this discussion, I did not look at the claims. I don't -- I couldn't answer that effectively right now. . . . Q. And so, as you sit here today, are you able to identify any difference between the claims of the '058 patent and what was in Solaris zones? A. I would have a difficult time being specific about claims because, as I said, I didn't review the claims. I can -- I am comfortable with differences in technology or differences in implementation, but the specific claims I would have to -- I couldn't answer right now.")

’058 Patent Claim 1	Disclosure
	• “Server consolidation, which allows multiple workloads to run on the same system, has become increasingly important as a way to improve the utilization of computing resources and reduce costs. Consolidation is common in mainframe environments, where technology to support running multiple workloads and even multiple operating systems on the same hardware has been evolving since the late 1960's. This technology is now becoming an important differentiator in the UNIX and Linux server market as well, both at the low end (virtual web hosting) and high end (traditional data center server consolidation). This paper introduces Solaris Zones (zones), a fully realized solution for server consolidation projects in a commercial UNIX operating system. By creating virtualized application execution environments within a single instance of the operating system, the facility strikes a unique balance between competing requirements. On the one hand, a system with multiple workloads needs to run those workloads in isolation, to ensure that applications can neither observe data from other applications nor affect their operation. It must also prevent applications from over-consuming system resources. On the other hand, the system as a whole has to be flexible, manageable, and observable, in order to reduce administrative costs and increase efficiency. By focusing on the support of multiple application environments rather than multiple operating system instances, zones meets isolation requirements without sacrificing manageability.” Price , D. and Tucker, A. (2004) *Solaris Zones: Operating System Support for Consolidating Commercial Workloads*. [https://www.researchgate.net/publication/220900659_Solaris_Zones_Operating_System_Support_for_Consolidating_Commercial_Workloads] • “Zones are a new operating system abstraction for partitioning systems, allowing multiple applications to run in isolation from each other on the same physical hardware. This isolation prevents processes running within a zone from monitoring or affecting processes running in other zones, seeing each other’s data, or manipulating the underlying hardware. Zones also provide an abstraction layer that separates applications from physical attributes of the machine on which they are deployed, such as physical device paths and network interface names.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • “These efforts differ from virtual machine implementations in that there is only one underlying operating system kernel, which is enhanced to provide increased isolation between groups of processes. The result is the ability to run multiple applications in isolation from each other within a single operating system instance.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][1], Chart A-2 (Tucker ‘556) at limitation 1[p][1], and Chart A-3 (Tucker Provisional) at limitation 1[p][1] as if fully set forth herein.

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 1	Disclosure
	To the extent Solaris Zones does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Solaris Zones. Further, one of ordinary skill in the art would have been motivated to modify Solaris Zones or combine it with any of the present prior art references found in Defendant’s Invalidity Contentions and any supplements thereto and the relevant section of charts for other prior art for the ’058 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Solaris Zones.

Chart B-24 **Invalidity Contentions: Solaris Zones**

Claim 2

’058 Patent Claim 2	Disclosure
2. A computing system as defined in claim 1, wherein in operation, multiple instances of an SLCSE stored in the shared library run simultaneously within the operating system.	Solaris Zones, as evidenced by the example citations below, discloses the computing system as defined in claim 1, wherein in operation, multiple instances of an SLCSE stored in the shared library run simultaneously within the operating system. *See, e.g.,* the disclosures set forth at claim elements 1[c] and 1[f][1]-[2]. *See also*, *e.g.*: • Rochette Tr. at 37:1-14 (“Q. Okay. So what was your understanding of Solaris zones? A. Oh gosh. That is a lengthy answer. Zones would provide a separate container, a separate security environment for each app and for our multiple -- and for an application to run it. So you could define a zone for each application you had running in Solaris. That zone would allow the application to have it -- potentially its own version of files. It would -- I don't recall if it had isolated networking, but it would provide a secure environment, and it would provide a separate instance of files and so forth.”) • Rochette Tr. at 83:22-88:8 ("Q. I think you mentioned earlier that Solaris zones had the capability of moving certain functionality that would normally be in an operating system kernel into a shared library; is that right? A. Yes, there were some capabilities that were represented outside of the kernel in user mode, correct. Q. What capabilities were those? A. File system separation. The use of something like the chroot command, or the chroot command you referred to earlier. The -- oh, I can't think of other examples. I mean, there were several examples of it. Q. And how did you learn that Solaris zones had that capability? A. Sitting down in front of a Solaris operating system and using it. Reading the documentations and using the capabilities and testing it. Q. And how did you determine that those capabilities were implemented in shared libraries rather than the kernel? A. If you go to an application and -- you would execute the command LDD. LDD shows you what libraries are required by a certain executable, and it would -- it's very clear -- clearly shown what libraries are referenced, and you could then look at the libraries and very clearly see what is implemented in those libraries. Q. So do you know if there was any shared library functionality in Solaris that was also in the kernel, in other words, the shared library functionality was a duplicate or performing a similar function to what was in the kernel? A. There were some networking capabilities and shared libraries and some -- for example, that would define what IP address is used, define what routing is used for a container separate from the kernel and was -- would be accessed by applications before going to the kernel. So you would have had a very similar capability in the kernel that was accessed by applications in these zones before going to the kernel. Q. So just for the sake of a concrete discussion, can you think of an example of a shared library that we could talk about that had this operating system functionality? A. I can't recall the names of them in Solaris. They would have been a lib.network.so or lib -- something with the name 'network' in it or 'net.' And it would have the network -- the things that would otherwise -- the things that were also present in a kernel, like defining an IP address, like defining routes, like defining -- not TCP protocol, but these would have been used by zones to configure the

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 2	Disclosure
	network capabilities present in that zone or container for use by the application. I don't remember the names. It's a long time ago, but there were shared libraries used by Solaris that did those things. Q. So for the networking shared library, if there were two applications using that shared library, would they each have their own instance in the sense of, you know, internal data used by the shared library having two different copies of that data? A. The way it works is the code for that library in question, once loaded into memory, is shared by all applications. That's the -- hence, the term 'shared library.' And -- so multiple zones could make use of the same networking library. The difference is that the data they use is distinct. So the actual code that gets executed is common across zones; the data that they use is separate for each zone. And any writes that would happen into memory . . . would implement an operating feature called copy-on-write, which would create a different instance of a page in memory, but it's specific to that zone.") • Rochette Tr. at 90:2-92:5 ("Q. And do you know if the claims of the '058 patent reflect that difference? A. When I reviewed the patents in preparation for this discussion, I did not look at the claims. I don't -- I couldn't answer that effectively right now. . . . Q. And so, as you sit here today, are you able to identify any difference between the claims of the '058 patent and what was in Solaris zones? A. I would have a difficult time being specific about claims because, as I said, I didn't review the claims. I can -- I am comfortable with differences in technology or differences in implementation, but the specific claims I would have to -- I couldn't answer right now.") • “Server consolidation, which allows multiple workloads to run on the same system, has become increasingly important as a way to improve the utilization of computing resources and reduce costs. Consolidation is common in mainframe environments, where technology to support running multiple workloads and even multiple operating systems on the same hardware has been evolving since the late 1960's. This technology is now becoming an important differentiator in the UNIX and Linux server market as well, both at the low end (virtual web hosting) and high end (traditional data center server consolidation). This paper introduces Solaris Zones (zones), a fully realized solution for server consolidation projects in a commercial UNIX operating system. By creating virtualized application execution environments within a single instance of the operating system, the facility strikes a unique balance between competing requirements. On the one hand, a system with multiple workloads needs to run those workloads in isolation, to ensure that applications can neither observe data from other applications nor affect their operation. It must also prevent applications from over-consuming system resources. On the other hand, the system as a whole has to be flexible, manageable, and observable, in order to reduce administrative costs and increase efficiency. By focusing on the support of multiple application environments rather than multiple operating system instances, zones meets isolation requirements without sacrificing manageability.” Price , D. and Tucker, A. (2004) *Solaris Zones: Operating System Support for Consolidating Commercial Workloads*. [https://www.researchgate.net/publication/220900659_Solaris_Zones_Operating_System_Support_for_Consolidating_Commercial_Workloads]

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 2	Disclosure
	• “Zones are a new operating system abstraction for partitioning systems, allowing multiple applications to run in isolation from each other on the same physical hardware. This isolation prevents processes running within a zone from monitoring or affecting processes running in other zones, seeing each other’s data, or manipulating the underlying hardware. Zones also provide an abstraction layer that separates applications from physical attributes of the machine on which they are deployed, such as physical device paths and network interface names.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • “These efforts differ from virtual machine implementations in that there is only one underlying operating system kernel, which is enhanced to provide increased isolation between groups of processes. The result is the ability to run multiple applications in isolation from each other within a single operating system instance.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][1], Chart A-2 (Tucker ‘556) at limitation 1[p][1], and Chart A-3 (Tucker Provisional) at limitation 1[p][1] as if fully set forth herein. To the extent Solaris Zones does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Solaris Zones. Further, one of ordinary skill in the art would have been motivated to modify Solaris Zones or combine it with any of the present prior art references found in Defendant’s Invalidity Contentions and any supplements thereto and the relevant section of charts for other prior art for the ’058 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Solaris Zones.

Chart B-24 **Invalidity Contentions: Solaris Zones**

Claim 3

’058 Patent Claim 3	Disclosure
3. A computing system according to claim 1 wherein OSCSEs corresponding to and capable of performing the same function as SLCSEs remain in the operating system kernel.	Solaris Zones, as evidenced by the example citations below, discloses the computing system according to claim 1 wherein OSCSEs corresponding to and capable of performing the same function as SLCSEs remain in the operating system kernel. *See, e.g.*, the disclosures set forth at claim element 1[b], 1[d][1]. *See also*, *e.g.*: • Rochette Tr. at 37:1-14 (“Q. Okay. So what was your understanding of Solaris zones? A. Oh gosh. That is a lengthy answer. Zones would provide a separate container, a separate security environment for each app and for our multiple -- and for an application to run it. So you could define a zone for each application you had running in Solaris. That zone would allow the application to have it -- potentially its own version of files. It would -- I don't recall if it had isolated networking, but it would provide a secure environment, and it would provide a separate instance of files and so forth.”) • Rochette Tr. at 83:22-88:8 ("Q. I think you mentioned earlier that Solaris zones had the capability of moving certain functionality that would normally be in an operating system kernel into a shared library; is that right? A. Yes, there were some capabilities that were represented outside of the kernel in user mode, correct. Q. What capabilities were those? A. File system separation. The use of something like the chroot command, or the chroot command you referred to earlier. The -- oh, I can't think of other examples. I mean, there were several examples of it. Q. And how did you learn that Solaris zones had that capability? A. Sitting down in front of a Solaris operating system and using it. Reading the documentations and using the capabilities and testing it. Q. And how did you determine that those capabilities were implemented in shared libraries rather than the kernel? A. If you go to an application and -- you would execute the command LDD. LDD shows you what libraries are required by a certain executable, and it would -- it's very clear -- clearly shown what libraries are referenced, and you could then look at the libraries and very clearly see what is implemented in those libraries. Q. So do you know if there was any shared library functionality in Solaris that was also in the kernel, in other words, the shared library functionality was a duplicate or performing a similar function to what was in the kernel? A. There were some networking capabilities and shared libraries and some -- for example, that would define what IP address is used, define what routing is used for a container separate from the kernel and was -- would be accessed by applications before going to the kernel. So you would have had a very similar capability in the kernel that was accessed by applications in these zones before going to the kernel. Q. So just for the sake of a concrete discussion, can you think of an example of a shared library that we could talk about that had this operating system functionality? A. I can't recall the names of them in Solaris. They would have been a lib.network.so or lib -- something with the name 'network' in it or 'net.' And it would have the network -- the things that would otherwise -- the things that were also present in a kernel, like defining an IP address, like defining routes, like defining -- not TCP protocol, but these would have been used by zones to configure the network capabilities present in that zone or container for use by the application. I don't remember the

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 3	Disclosure
	names. It's a long time ago, but there were shared libraries used by Solaris that did those things. Q. So for the networking shared library, if there were two applications using that shared library, would they each have their own instance in the sense of, you know, internal data used by the shared library having two different copies of that data? A. The way it works is the code for that library in question, once loaded into memory, is shared by all applications. That's the -- hence, the term 'shared library.' And -- so multiple zones could make use of the same networking library. The difference is that the data they use is distinct. So the actual code that gets executed is common across zones; the data that they use is separate for each zone. And any writes that would happen into memory . . . would implement an operating feature called copy-on-write, which would create a different instance of a page in memory, but it's specific to that zone.") • Rochette Tr. at 90:2-92:5 ("Q. And do you know if the claims of the '058 patent reflect that difference? A. When I reviewed the patents in preparation for this discussion, I did not look at the claims. I don't -- I couldn't answer that effectively right now. . . . Q. And so, as you sit here today, are you able to identify any difference between the claims of the '058 patent and what was in Solaris zones? A. I would have a difficult time being specific about claims because, as I said, I didn't review the claims. I can -- I am comfortable with differences in technology or differences in implementation, but the specific claims I would have to -- I couldn't answer right now.") • “Server consolidation, which allows multiple workloads to run on the same system, has become increasingly important as a way to improve the utilization of computing resources and reduce costs. Consolidation is common in mainframe environments, where technology to support running multiple workloads and even multiple operating systems on the same hardware has been evolving since the late 1960's. This technology is now becoming an important differentiator in the UNIX and Linux server market as well, both at the low end (virtual web hosting) and high end (traditional data center server consolidation). This paper introduces Solaris Zones (zones), a fully realized solution for server consolidation projects in a commercial UNIX operating system. By creating virtualized application execution environments within a single instance of the operating system, the facility strikes a unique balance between competing requirements. On the one hand, a system with multiple workloads needs to run those workloads in isolation, to ensure that applications can neither observe data from other applications nor affect their operation. It must also prevent applications from over-consuming system resources. On the other hand, the system as a whole has to be flexible, manageable, and observable, in order to reduce administrative costs and increase efficiency. By focusing on the support of multiple application environments rather than multiple operating system instances, zones meets isolation requirements without sacrificing manageability.” Price , D. and Tucker, A. (2004) *Solaris Zones: Operating System Support for Consolidating Commercial Workloads*. [https://www.researchgate.net/publication/220900659_Solaris_Zones_Operating_System_Support_for_Consolidating_Commercial_Workloads]

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 3	Disclosure
	• “Zones are a new operating system abstraction for partitioning systems, allowing multiple applications to run in isolation from each other on the same physical hardware. This isolation prevents processes running within a zone from monitoring or affecting processes running in other zones, seeing each other’s data, or manipulating the underlying hardware. Zones also provide an abstraction layer that separates applications from physical attributes of the machine on which they are deployed, such as physical device paths and network interface names.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • “These efforts differ from virtual machine implementations in that there is only one underlying operating system kernel, which is enhanced to provide increased isolation between groups of processes. The result is the ability to run multiple applications in isolation from each other within a single operating system instance.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][1], Chart A-2 (Tucker ‘556) at limitation 1[p][1], and Chart A-3 (Tucker Provisional) at limitation 1[p][1] as if fully set forth herein. To the extent Solaris Zones does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Solaris Zones. Further, one of ordinary skill in the art would have been motivated to modify Solaris Zones or combine it with any of the present prior art references found in Defendant’s Invalidity Contentions and any supplements thereto and the relevant section of charts for other prior art for the ’058 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Solaris Zones.

Chart B-24 **Invalidity Contentions: Solaris Zones**

Claim 4

’058 Patent Claim 4	Disclosure
4. A computing system according to claim 1 wherein the one or more SLCSEs provided to one of the plurality of software applications having exclusive use thereof, use system calls to access services in the operating system kernel.	Solaris Zones, as evidenced by the example citations below, discloses the computing system according to claim 1 wherein the one or more SLCSEs provided to one of the plurality of software applications having exclusive use thereof, use system calls to access services in the operating system kernel. *See, e.g.*, • Rochette Tr. at 37:1-14 (“Q. Okay. So what was your understanding of Solaris zones? A. Oh gosh. That is a lengthy answer. Zones would provide a separate container, a separate security environment for each app and for our multiple -- and for an application to run it. So you could define a zone for each application you had running in Solaris. That zone would allow the application to have it -- potentially its own version of files. It would -- I don't recall if it had isolated networking, but it would provide a secure environment, and it would provide a separate instance of files and so forth.”) • Rochette Tr. at 83:22-88:8 ("Q. I think you mentioned earlier that Solaris zones had the capability of moving certain functionality that would normally be in an operating system kernel into a shared library; is that right? A. Yes, there were some capabilities that were represented outside of the kernel in user mode, correct. Q. What capabilities were those? A. File system separation. The use of something like the chroot command, or the chroot command you referred to earlier. The -- oh, I can't think of other examples. I mean, there were several examples of it. Q. And how did you learn that Solaris zones had that capability? A. Sitting down in front of a Solaris operating system and using it. Reading the documentations and using the capabilities and testing it. Q. And how did you determine that those capabilities were implemented in shared libraries rather than the kernel? A. If you go to an application and -- you would execute the command LDD. LDD shows you what libraries are required by a certain executable, and it would -- it's very clear -- clearly shown what libraries are referenced, and you could then look at the libraries and very clearly see what is implemented in those libraries. Q. So do you know if there was any shared library functionality in Solaris that was also in the kernel, in other words, the shared library functionality was a duplicate or performing a similar function to what was in the kernel? A. There were some networking capabilities and shared libraries and some -- for example, that would define what IP address is used, define what routing is used for a container separate from the kernel and was -- would be accessed by applications before going to the kernel. So you would have had a very similar capability in the kernel that was accessed by applications in these zones before going to the kernel. Q. So just for the sake of a concrete discussion, can you think of an example of a shared library that we could talk about that had this operating system functionality? A. I can't recall the names of them in Solaris. They would have been a lib.network.so or lib -- something with the name 'network' in it or 'net.' And it would have the network -- the things that would otherwise -- the things that were also present in a kernel, like defining an IP address, like defining routes, like defining -- not TCP protocol, but these would have been used by zones to configure the network capabilities present in that zone or container for use by the application. I don't remember the

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 4	Disclosure
	names. It's a long time ago, but there were shared libraries used by Solaris that did those things. Q. So for the networking shared library, if there were two applications using that shared library, would they each have their own instance in the sense of, you know, internal data used by the shared library having two different copies of that data? A. The way it works is the code for that library in question, once loaded into memory, is shared by all applications. That's the -- hence, the term 'shared library.' And -- so multiple zones could make use of the same networking library. The difference is that the data they use is distinct. So the actual code that gets executed is common across zones; the data that they use is separate for each zone. And any writes that would happen into memory . . . would implement an operating feature called copy-on-write, which would create a different instance of a page in memory, but it's specific to that zone.") • Rochette Tr. at 90:2-92:5 ("Q. And do you know if the claims of the '058 patent reflect that difference? A. When I reviewed the patents in preparation for this discussion, I did not look at the claims. I don't -- I couldn't answer that effectively right now. . . . Q. And so, as you sit here today, are you able to identify any difference between the claims of the '058 patent and what was in Solaris zones? A. I would have a difficult time being specific about claims because, as I said, I didn't review the claims. I can -- I am comfortable with differences in technology or differences in implementation, but the specific claims I would have to -- I couldn't answer right now.") • “Server consolidation, which allows multiple workloads to run on the same system, has become increasingly important as a way to improve the utilization of computing resources and reduce costs. Consolidation is common in mainframe environments, where technology to support running multiple workloads and even multiple operating systems on the same hardware has been evolving since the late 1960's. This technology is now becoming an important differentiator in the UNIX and Linux server market as well, both at the low end (virtual web hosting) and high end (traditional data center server consolidation). This paper introduces Solaris Zones (zones), a fully realized solution for server consolidation projects in a commercial UNIX operating system. By creating virtualized application execution environments within a single instance of the operating system, the facility strikes a unique balance between competing requirements. On the one hand, a system with multiple workloads needs to run those workloads in isolation, to ensure that applications can neither observe data from other applications nor affect their operation. It must also prevent applications from over-consuming system resources. On the other hand, the system as a whole has to be flexible, manageable, and observable, in order to reduce administrative costs and increase efficiency. By focusing on the support of multiple application environments rather than multiple operating system instances, zones meets isolation requirements without sacrificing manageability.” Price , D. and Tucker, A. (2004) *Solaris Zones: Operating System Support for Consolidating Commercial Workloads*. [https://www.researchgate.net/publication/220900659_Solaris_Zones_Operating_System_Support_for_Consolidating_Commercial_Workloads]

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 4	Disclosure
	• “Zones are a new operating system abstraction for partitioning systems, allowing multiple applications to run in isolation from each other on the same physical hardware. This isolation prevents processes running within a zone from monitoring or affecting processes running in other zones, seeing each other’s data, or manipulating the underlying hardware. Zones also provide an abstraction layer that separates applications from physical attributes of the machine on which they are deployed, such as physical device paths and network interface names.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • “These efforts differ from virtual machine implementations in that there is only one underlying operating system kernel, which is enhanced to provide increased isolation between groups of processes. The result is the ability to run multiple applications in isolation from each other within a single operating system instance.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][1], Chart A-2 (Tucker ‘556) at limitation 1[p][1], and Chart A-3 (Tucker Provisional) at limitation 1[p][1] as if fully set forth herein. To the extent Solaris Zones does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Solaris Zones. Further, one of ordinary skill in the art would have been motivated to modify Solaris Zones or combine it with any of the present prior art references found in Defendant’s Invalidity Contentions and any supplements thereto and the relevant section of charts for other prior art for the ’058 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Solaris Zones.

Chart B-24 **Invalidity Contentions: Solaris Zones**

Claim 18

’058 Patent Claim 18	Disclosure
18. A computer system as defined in claim 2 wherein SLCSEs are not copies of OSCSEs.	Solaris Zones, as evidenced by the example citations below, discloses the computer system as defined in claim 2 wherein SLCSEs are not copies of OSCSEs. *See, e.g.,*: • Rochette Tr. at 37:1-14 (“Q. Okay. So what was your understanding of Solaris zones? A. Oh gosh. That is a lengthy answer. Zones would provide a separate container, a separate security environment for each app and for our multiple -- and for an application to run it. So you could define a zone for each application you had running in Solaris. That zone would allow the application to have it -- potentially its own version of files. It would -- I don't recall if it had isolated networking, but it would provide a secure environment, and it would provide a separate instance of files and so forth.”) • Rochette Tr. at 83:22-88:8 ("Q. I think you mentioned earlier that Solaris zones had the capability of moving certain functionality that would normally be in an operating system kernel into a shared library; is that right? A. Yes, there were some capabilities that were represented outside of the kernel in user mode, correct. Q. What capabilities were those? A. File system separation. The use of something like the chroot command, or the chroot command you referred to earlier. The -- oh, I can't think of other examples. I mean, there were several examples of it. Q. And how did you learn that Solaris zones had that capability? A. Sitting down in front of a Solaris operating system and using it. Reading the documentations and using the capabilities and testing it. Q. And how did you determine that those capabilities were implemented in shared libraries rather than the kernel? A. If you go to an application and -- you would execute the command LDD. LDD shows you what libraries are required by a certain executable, and it would -- it's very clear -- clearly shown what libraries are referenced, and you could then look at the libraries and very clearly see what is implemented in those libraries. Q. So do you know if there was any shared library functionality in Solaris that was also in the kernel, in other words, the shared library functionality was a duplicate or performing a similar function to what was in the kernel? A. There were some networking capabilities and shared libraries and some -- for example, that would define what IP address is used, define what routing is used for a container separate from the kernel and was -- would be accessed by applications before going to the kernel. So you would have had a very similar capability in the kernel that was accessed by applications in these zones before going to the kernel. Q. So just for the sake of a concrete discussion, can you think of an example of a shared library that we could talk about that had this operating system functionality? A. I can't recall the names of them in Solaris. They would have been a lib.network.so or lib -- something with the name 'network' in it or 'net.' And it would have the network -- the things that would otherwise -- the things that were also present in a kernel, like defining an IP address, like defining routes, like defining -- not TCP protocol, but these would have been used by zones to configure the network capabilities present in that zone or container for use by the application. I don't remember the names. It's a long time ago, but there were shared libraries used by Solaris that did those things. Q. So for

Chart B-24 **Invalidity Contentions: Solaris Zones**

'058 Patent Claim 18	Disclosure
	the networking shared library, if there were two applications using that shared library, would they each have their own instance in the sense of, you know, internal data used by the shared library having two different copies of that data? A. The way it works is the code for that library in question, once loaded into memory, is shared by all applications. That's the -- hence, the term 'shared library.' And -- so multiple zones could make use of the same networking library. The difference is that the data they use is distinct. So the actual code that gets executed is common across zones; the data that they use is separate for each zone. And any writes that would happen into memory . . . would implement an operating feature called copy-on-write, which would create a different instance of a page in memory, but it's specific to that zone.") • Rochette Tr. at 90:2-92:5 ("Q. And do you know if the claims of the '058 patent reflect that difference? A. When I reviewed the patents in preparation for this discussion, I did not look at the claims. I don't -- I couldn't answer that effectively right now. . . . Q. And so, as you sit here today, are you able to identify any difference between the claims of the '058 patent and what was in Solaris zones? A. I would have a difficult time being specific about claims because, as I said, I didn't review the claims. I can -- I am comfortable with differences in technology or differences in implementation, but the specific claims I would have to -- I couldn't answer right now.") • "Server consolidation, which allows multiple workloads to run on the same system, has become increasingly important as a way to improve the utilization of computing resources and reduce costs. Consolidation is common in mainframe environments, where technology to support running multiple workloads and even multiple operating systems on the same hardware has been evolving since the late 1960's. This technology is now becoming an important differentiator in the UNIX and Linux server market as well, both at the low end (virtual web hosting) and high end (traditional data center server consolidation). This paper introduces Solaris Zones (zones), a fully realized solution for server consolidation projects in a commercial UNIX operating system. By creating virtualized application execution environments within a single instance of the operating system, the facility strikes a unique balance between competing requirements. On the one hand, a system with multiple workloads needs to run those workloads in isolation, to ensure that applications can neither observe data from other applications nor affect their operation. It must also prevent applications from over-consuming system resources. On the other hand, the system as a whole has to be flexible, manageable, and observable, in order to reduce administrative costs and increase efficiency. By focusing on the support of multiple application environments rather than multiple operating system instances, zones meets isolation requirements without sacrificing manageability." Price , D. and Tucker, A. (2004) *Solaris Zones: Operating System Support for Consolidating Commercial Workloads*. [https://www.researchgate.net/publication/220900659_Solaris_Zones_Operating_System_Support_for_Consolidating_Commercial_Workloads] • "Zones are a new operating system abstraction for partitioning systems, allowing multiple applications to run in isolation from each other on the same physical hardware. This isolation prevents processes running

Chart B-24 **Invalidity Contentions: Solaris Zones**

’058 Patent Claim 18	Disclosure
	within a zone from monitoring or affecting processes running in other zones, seeing each other’s data, or manipulating the underlying hardware. Zones also provide an abstraction layer that separates applications from physical attributes of the machine on which they are deployed, such as physical device paths and network interface names.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • “These efforts differ from virtual machine implementations in that there is only one underlying operating system kernel, which is enhanced to provide increased isolation between groups of processes. The result is the ability to run multiple applications in isolation from each other within a single operating system instance.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][1], Chart A-2 (Tucker ‘556) at limitation 1[p][1], and Chart A-3 (Tucker Provisional) at limitation 1[p][1] as if fully set forth herein. To the extent Solaris Zones does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Solaris Zones. Further, one of ordinary skill in the art would have been motivated to modify Solaris Zones or combine it with any of the present prior art references found in Defendant’s Invalidity Contentions and any supplements thereto and the relevant section of charts for other prior art for the ’058 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Solaris Zones.